

开发月刊

Development Monthly

2011年6月
总第003期

向Java开战？

最受Web前端开发者欢迎

五大开发工具

当Java遇见了Objective-C

目录

2011.06

contents

编程排行 Billboard

- 3 六月编程语言排行榜：为孩子创造的语言Scratch

专题报道 《欢迎Java 7》

- 6 51CTO开发频道寄语
7 再探Java 7新特性，代码及性能测试
10 Java 7，技术标准商业咒语
12 向Java开战?别搞错了对象
14 相同中不同:Java程序员不应低看C#

技术热点 Techlogy hot

- 18 新手软件项目经理：你知道的太多了
21 开源NoSQL系统又发力，Membase 1.7发布
22 新里程碑到来 开启PHP框架新时代
25 最受Web开发者欢迎的五大工具

移动开发 Mobile Develope

- 28 当Java遇见了Objective-C
33 Office应用 移动商务开发的新航标

特别推荐 Best topic

- 35 见微知著 大型网站PHP开发之道



51CTO.com
技术成就梦想

出版方: 北京无忧创想信息技术有限公司

责任编辑: 边艾婧

联系方法: bianaj@51cto.com 010-68476606 (分机 8094)

出版日期: 2011 年 6 月 15 日

6月编程排行榜:为孩子创造Scratch

Tiobe 发布了新一期编程语言排行榜。涨势方面基本与上月相似,值得一说的是 Lua,它是首次进入前 10 名,不知道这是否与苹果允许 Lua 运行在 IOS 系统上有关。反观传统的 Web 编程语言正在经历一个艰难的时期,它们正在迅速丢失市场,看看排行榜中的 PHP、Ruby 和 Python 的下降就知道了。

端午小长假前后, Tiobe 发布了新一期编程语言排行榜。涨势方面基本与上月相似,值得一说的是 Lua,它是首次进入前 10 名,不知道这是否与苹果允许 Lua 运行在 IOS 系统上有关,包括时下很流行的愤怒的小鸟也被写到 Lua 中。反观传统的 Web 语言,他们正在迅速失去目前的市场份额。下面是 2011 年 6 月的编程语言排行榜榜单。

Position Jun 2011	Position Jun 2010	Delta in Position	Programming Language
1	2	↑	JAVA
2	1	↓	C
3	3	=	C++
4	5	↑↑	C#
5	4	↓	PHP
6	5	↓	(Visual) Basic
7	10	↑↑↑	Objective-C
8	7	↓	Python
9	8	↓	Perl
10	20	↑↑↑↑↑↑↑↑	Lua
11	12	↑	JavaScript
12	11	↓	Ruby
13	9	↓↓↓	Delphi/Object Pascal
14	10	↑↑	Lisp
15	15	=	Pascal
16		=	Assembly*
17	21	↑↑↑↑	Transact-SQL
18	25	↑↑↑↑↑↑	RPG (OS/400)
19	23	↑↑↑↑	Ada
20		=	Scheme*

什么是 Scratch ?

Scratch 是一款由麻省理工学院 (MIT) 设计开发的一款面向儿童的简易编程工具。针对 8 岁以上孩子们的认知水平,以及对于界面的喜好, MIT 做了相当深入研究和颇具针对性的设计开发。不仅易于孩子们使用,又能寓教于乐,让孩子们获得创作中的乐趣。Scratch 的下载和使用是完全免费的,开发了 Windows 系统,苹果系统, Linux 系统下运行的版本。

Scratch 1.4 Download



Scratch Installer For Mac OS X

Compatible with Mac OS X 10.4 or later

MacScratch1.4.dmg



Scratch Installer for Windows

Compatible with Windows 2000, XP, Vista, and 7

ScratchInstaller1.4.exe

See below for additional Windows options



Scratch Installer for Ubuntu

Compatible with Ubuntu version 9.04 and later

Scratch_1.4.0.1-0ubuntu5_i386.deb

See the Scratch on Linux page for more informat

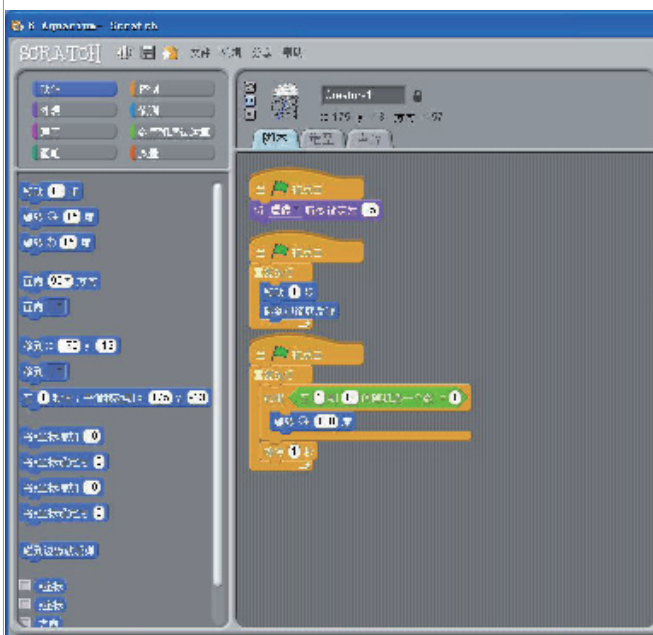
6 月编程语言排行榜：为孩子创造的语言 Scratch II

【编者碎语】这个面向小朋友的软件可爱的界面深深的吸引了小编我,于是与官网尝试着去下载,发现下载很简单。

只需要简单的填写一下自己的信息,就可以转到下载页面,而且就连下载页面也是做得相当的可爱。碎语不多说,小编现在要去尝试着用一用这个可爱软件了。

软件的特点:

- ◆使用者可以不认识英文单词,也可以不会使用键盘。
- ◆构成程序的命令和参数通过积木形状的实现。
- ◆用鼠标拖动模块到程序编辑栏就可以了。
- ◆中间的黄色部分是编辑好的程序代码,左边是可以用来选择的功能模块,右边上部是程序预览和运行窗口,右边下部是角色窗口。



刚刚尝试着用了一下这个软件,这真是一个很可爱的软件,有趣的模块形式,就像是在搭建积

木。你能想象这是一个小孩子做的拖拽游戏吗?

程序模块分为 8 个大类

软件是多语言版本,根据操作系统,会自动改成中文界面。积木模块包括 8 个大类,100 多个功能。包括了过程一个完整程序的每个环节,甚至数组。在原版中是没有函数调用的。也就是说复杂的功能用重复编写相同的代码。在自由软件开发组织中有人进一步开发了制作自定义积木(自己编函数)的功能。



学习方法

孩子不用记住命令不代表不需要指导命令。积木模块包括 8 个大类,100 多个功能。包括了过程一个完整程序的每个环节,甚至数组和函数。这些图形化的积木使用需要老师或者家长正确引导。孩子从模仿的过程中,能感悟到条件语句,循环语句,判断。能理解参数和命令的区别。让孩子理性的总结出来是很困难的,引导是必须的。

第一届 Scratch 大会

2008 年 7 月 24-26 日, MIT 的 Scratch 团队举办了第一次的全球 Scratch 社群聚会「Scratch@MIT」,约有 300 位老师、研究学者、软体开发者前来参与。该研讨会的目的是:

- ◆研讨孩子们用 Scratch 学会了些什么。

6 月编程语言排行榜：为孩子创造的语言 Scratch III

◆ 分享 Scratch 是如何在家里、教室、社区中心被使用的故事。

◆ 参与要亲自动手操作的工作坊，学习 Scratch 的新技巧和策略。

◆ 和 MIT 的 Scratch 团队一起探索 Scratch 的未来发展方向。

【小编碎语】

这个软件的优点是将程式设计变得很容易，只要使用简易的鼠标拖拉方式，将指令从程式指令区移进程序脚本区即可。很容易上手，官方网站上有相应的教程，界面也很可爱，吸引人，版本支持多国语言。Scratch 程序设计的交互性特别强，在程序设计中只要用鼠标双击任何语句块，以及语句块的组合，都可以马上执行看执行效果。

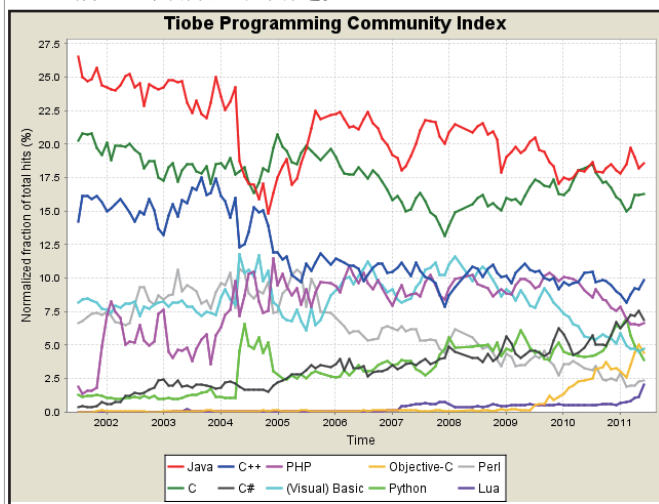
官网地址：<http://scratch.mit.edu/>

下载地址：<http://scratch.mit.edu/download/>

有优点，就会有缺点，缺点在于对象多是位图，放大后会有马赛克现象，美观上有所欠缺吧。

下面是本期编程语言排行榜的其他排名数据和趋势走向。

前 10 名编程语言走势图



下面是第 50 到 100 的编程语言排名

NATURAL, COBOL, PL/I, Clean, Haskell, APL, OpenEdge ABL, Smalltalk, Scala, J, Ladder Logic, (Visual) FoxPro, 4th Dimension/4D, ABC, Algol, Arc, ATLAS, Avenue, Awk, Bash, bc, BETA, BlitzMax, Boo, Bourne shell, CFML, cg, CL (OS/400), cT, Dylan, Eiffel, Erlang, Factor, Groovy, Icon, IDL, Inform, Io, Korn shell, LabVIEW, Limbo, MAD, Magic, Maple, Mercury, Monkey, MUMPS, Oberon, OpenCL, Oz, PILOT, PowerShell, Prolog, Revolution, S, SIGNAL, Standard ML, Tcl, TOM, VBScript, VHDL

下面给出了编程语言类别的一年变化趋势

Category	Ratings Jun 2011	Delta Jun 2010
Object-Oriented Languages	56.5%	+2.5%
Procedural Languages	37.4%	-3.9%
Functional Languages	4.6%	+1.4%
Logical Languages	1.5%	-0.1%

Category	Ratings Jun 2011	Delta Jun 2010
Statically Typed Languages	64.8%	+2.2%
Dynamically Typed Languages	35.2%	-2.2%

原文链接：

<http://developer.51cto.com/art/201106/267258.htm>

51CTO 编程语言排行榜

编程语言排行榜是 51CTO 开发频道根据 Tiobe 编程语言排行编辑整理而成，每月一期，记录当月编程语言的亮点：这里既有王者 Java，也有风靡一时的 Delphi。我们试图通过对某一门编程语言的深度分析，透视技术和软件业的兴衰，把握流行的脉络和发展，为您提供有价值的趋势参考。

Java 7 and Polyglot Programming on the JVM

The Well-Grounded Java Developer



Java 7, 是 JDK 7 的另一个常用称呼, 也叫做 Java SE 7。JDK 包括 Java 语言 API 类库, Java 虚拟机 JVM 以及 Java 运行时 JRE。根据 2009 年底的 Devoxx 大会, Java 7 会在 2010 年秋季正式到来, 其中包括很多对平台意义重大的新特性, 不过习惯跳票的 Java 7 还是延后到 2011 年。

专题: Java 7, 下一代 Java 开发技术详解

<http://developer.51cto.com/art/201001/175830.htm>

2011 年 6 月 8 日, Java 7(JSR 336) 通过了公开评审投票。共 16 家公司参与投票, 只有 Google 一家反对, 投票通过。

究竟 Java 7 是什么样子? 我们将带您走进 Java 7 的世界。

51CTO 开发频道寄语

■ 编者按

今天这篇文章将包括 Java 7 的最新更新,代码示例和对之前版本 Java 的性能比较等。我们可以发现,Java 7 在语言、内核和客户端方面的改进和提升相当明显。

再探Java 7新特性, 代码及性能测试

如果说今年 Java 开发平台有什么大事儿,那就要数将于年底发布的 Java 7 了。之前 51CTO 曾报道过 Java 7 已经完成的七大新功能,包括新增的垃圾回收器 G1、JVM 动态语言支持以及模块化等等。

进入 2010 年,Java 7 有哪些新的进展? 这些新的特性更新对我们有哪些好处? 该如何使用? Java 7 性能如何? 本文将给你答案。

这里的内容主要包括其最新更新、一些代码实例、跟以前版本的 Java 进行性能比较需要参考的一些基准、以及什么时候它才会发布等信息。

首先,让我们来看最重要的事情。为了确定 JDK 7 中有哪些语言方面的微小变化,有关人员建立了一个名叫 Project Coin 的项目。描述了最终的五个变化(比五个多一点)。

Java 7 更新内容如下所示:

- ◆ 允许在 switch 中使用字符串
- ◆ 自动资源管理
- ◆ 通用实例创建来改进类型推断(diamond)
- ◆ 简化的 Varargs 方法调用
- ◆ 更好的整型文字综合性建议
- ◆ Collections 集合的语言支持
- ◆ JSR292 的语言支持

在 the OpenJDK 7 特性页面中,你还可以看到

其他的功能。这些功能分为不同的种类:

- ◆ 虚拟机(VM)
- ◆ 压缩的 64 位对象指针
- ◆ G1 垃圾回收器 GC (G1)
- ◆ JSR 292: 非 Java 语言的虚拟机支持 (动态调用, InvokeDynamic)

语言方面(lang)

- ◆ SR 294: 模块化编程语言以及虚拟机支持
- ◆ JSR 308: Java 类型注释
- ◆ 语言微小增强(我所谈论的 Project Coin)
- ◆ JSR TBD: Project Lambda

内核(core)

- ◆ 模块化(Jigsaw 项目)
- ◆ 对类加载器的结构进行升级
- ◆ 关闭 URLClassLoader 的方法
- ◆ Unicode 5.1 标准
- ◆ 并行以及集合的升级(jsr166y)
- ◆ JSR 203: Java 平台(NIO.2)的更多新型

I/O API

- ◆ SCTP (流控制传输协议, Stream Control Transmission Protocol)
- ◆ SDP (套接字直接协议, Sockets Direct Protocol)
- ◆ 椭圆曲线加密技术(ECC)

再探 Java 7 新特性,代码示例及性能测试 II

作者 / 阿风

客户端(client)

- ◆ Java 2D 的 XRender 管道
- ◆ 转发端口 6u10 部署特性
- ◆ 为 6u10 图形功能创建新的平台 API
- ◆ Swing 的 Nimbus 外观和感觉
- ◆ Swing 的 JLayer 元件

网络(web)

- ◆ 更新 XML 的栈

就像你所看到的,这涉及了很多东西。几个月前,我亲自尝试了新的 Garbage Collector (GC),其性能表现给我留下了非常深刻的印象。不幸的是,JVM 几个小时就会崩溃一次,所以这个产品不能使用 JVM。虽然在 Java 1.6 中也可以使用这个 GC,但也会出现同样的问题,经常会出现崩溃。

我想,这就是 Java1.7 增加新特性的原因。那么,现在我们最好去看一些代码实例。

Java 7 新特性的代码实例

下面列出的大多数例子都是来源于 Joe Wright 博客中(Java7 中的新语言特性)的高水平文章。

集合的语言支持

在这里我们主要讲的是,当创建一个 List、Set 或者 Map 的时候,你怎样尽量少写代码。你不必先实例化 Object,然后再给 Collection 添加元素。你现在只需 1 行代码就可以完成。

```
List list = ["item"]; String item = list[0];

Set set = {"item"};

Map map = {"key" : 1};

int value = map["key"];
```

自动资源管理

由于 try / catch 语句的原因,冗长的代码令人非常头痛。你或许会喜欢这个全新的特性。

实际上,下面这些代码:

```
BufferedReader br = new
BufferedReader(new FileReader(path));

try { return br.readLine(); }

finally { br.close(); }
```

转变成了如下这种形式:

```
try (BufferedReader br = new
BufferedReader(new FileReader(path))) {

return br.readLine(); }
```

通用实例创建来改进类型推断(diamond)

当你把对象的接口申明指定成范型后,你在对象实例化时不得不再指定一次。现在,你不必如此了,因为你可以这样:

```
Map<> map = new HashMap<>();
```

数值文字的加强

我不敢肯定这个对大多数人都有用。你可以这样做:

```
int billion = 1_000_000_000;
```

允许在 switch 中使用字符串

这个无需解释,其意思很明确。

```
String availability = "available";
switch(availability) { case "available": //code
break; case "unavailable":
//code break;
case "merged": //code
default: //code break; }
```


再探 Java 7 新特性,代码示例及性能测试 III

作者 / 阿风

二进制文字

你可以使用前缀 0b 创建二进制文字

```
int binary = 0b1001_1001;
```

以上这些就是 Java1.7 的代码实例。如果有人能给我指出还有哪些没有包含进去,那就更好了。我敢肯定,已经有其他的开发人员对此进行了关注。

Java 1.7 的性能表现

Java 7 的性能有多大的提升? 这里我们来针对 Java 7 做一个测试,内容如下。

在一台装有 ArchLinux 系统的 Macbook Pro 电脑上(英特尔 Duo CPU T7700,主频 2.40GHz,有两年的使用时间)运行了这些测试。

内存是 2Gb 的,把 Heap Size 设置成了 728m(-Xms728m -Xmx728m)。

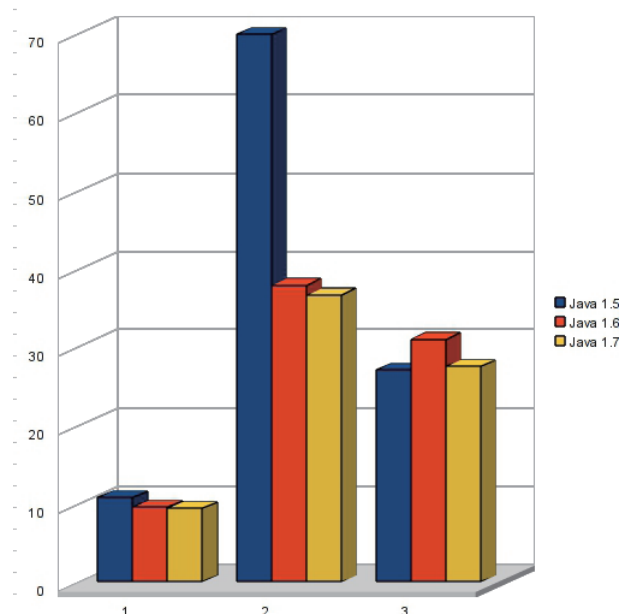
◆测试 1 为一个 List 添加 100 万个字符串值(String 字符串是一个 UUID,是用 UUID.randomUUID() 产生的)。

◆测试 2 带有 100 万键、值对的 HashMap。每个键、值对通过并行线程进行计算。键是一个 UUID,值 int 是用 Math.random() 产生的。

◆测试 3 把 100 万个 ArrayList 条目打印到一定数量的文件(1000 个)中。把条目写进恰巧并行的不同文件中。

我只比较了 Java1.6 (1.6.0_19) 和 Java 1.7 (b87)。后来根据评论的要求,我把 Java1.5 也添加了进来,但是并没有添加 Java1.4,因为它的时间太久了。

相关结果如下所示:



	Java 1.5	Java 1.6	Java 1.7
Test 1	10,698 秒	9,481 秒	9,328 秒
Test 2	69,827 秒	37,935 秒	36,636 秒
Test 3	26,931 秒	30,868 秒	27,383 秒

显然,不同版本之间性能差异并不特别大。

Java 7 的发布日期

在 2009 年 11 月的时候,Java1.7 计划是在 2010 年 9 月发布,到那时还将发布 3 个里程碑版本。其中,里程碑 6 在 build 84 版中已经完成。

里程碑 7 的第一个测试版 B85 也计划在 2010 年 3 月 4 日完成,而本文使用的 B87 版本已在 2010 年 3 月 25 日发布。这样看起来,Java 1.7 很有可能在 2010 年 9 月发布。让我们等待 Java 7 的到来吧。

Java 7，技术标准商业咒语

Java 7 技术规范背后的商业纷争像咒语一样影响着 Java 的技术进程,对持有技术标准的厂商而言 Java 却是商业利益和竞争的砝码。



我们一直在关注 Java 7,对于 Java 这样占据软件开发领域半壁江山的技术平台,没人会忽略它的新版本。但对于一门技术,特别是 IT 技术,使用者和拥有者却有截然不同的理解。

开发者的期望往往是更高效、更方便,更贴合当前应用场景的进化需求,也就是提高软件开发者的生产力。而对于持有技术标准的厂商而言,Java 要比其自身技术的结构、语法和特性要复杂得多,是商业利益和竞争的砝码。一再延期的 Java 7 就是这样一个被施了商业咒语的技术标准。

在进一步说明之前,我们必须明确 Java 标准的两个重要组成部分——JCP 和 JSR。JCP (Java Community Process) 是一个类似 ISO 标准委员会的组织,Java 的一切标准化工作由 JCP 负责。JSR (Java Specification Requests) 是 Java 请求规范,每一个 JSR 规范都包括一些 Java 技术细则、参考实现 (RI) 和一个兼容性测试工具 (TCK) 组成。Java 标准的每一项新的功能组成由 JSR 来封装后递交 JCP 审核。

如果某个社区或厂商希望 Java 增加一项功能,那就需要向 JCP 提交 JSR 并等待 JCP 的执行委员会投票通过并建立这个 Java 规范。51CTO 之前所报道的《Java 7 已经完成的七大新功能预览》中所提到的新功能都是通过 JSR 提交并得到 JCP 同意之后才实现的。

对于 Java 的发展而言,JCP 的权利极大,它可以决定一项 Java 规范是否会出现下一个版本的 Java 标准中,也可以决定下一个标准对怎样的应用开发更有利。

这其中的关键就是,JCP 被谁所控制。JCP 由 Sun 一手创建的,用来发展和更新 Java 技术规范、参考实现 (RI)、技术兼容包 (TCK),Sun 被 Oracle 收购后,JCP 也由 Oracle 接手。

事实上,在 Oracle 收购 Sun 之前,众多的 Java 开发者及已经开始纷纷逃离 JCP 而转换到一些小型的论坛。大批逃离者称,JCP 已经失去了它的吸引力,它不再是被当成响应和解决 Java 开发者问题的社区,相反,JCP 已经成为被一些大公司所利用 JSP 通过的橡皮图章。

这些大公司凭着自身的影响和力量来指定各种规范从而达到满足自己的利益。整个 JCP 更像是一个商业擂台,多位法师念出自己的咒语试图在 Java 技术的发展中对自己有利。

这一切在 Oracle 接手 Sun 之后变得更糟。在 Sun 和 Apache 基金会之间,关于 JDK 规范应用范围的纷争由来已久。之前 Sun 考虑到 Java ME 在授权方面的收入,不愿意为 Apache 提供一个公平的竞争环境。

目前大热的 Google Android 系统之所以会选择 Dalvik 虚拟机也是这个原因。类似的商业

Java 7, 技术标准商业咒语 II

博弈和争论在 Oracle 接手 JCP 后没有改变, 只是变得更为直接。

目前, Java 7 的 JSR 的官方规范还从未被 Sun 或 Oracle 提交给 JCP。Sun 的工程师还没有启动 OpenJDK JDK 7 中的 Java 7 功能规划。恰恰在这段时间, 我们听到各种技术项目的决策(不引入闭包, 引入闭包, 等等)。目前, 大部分已有的功能还处于功能级别以下(under feature-level specs), 所以, 我们看到 51CTO.com 上周所报道的新闻《Java 7 未按时发布 计划再次延期》。

注: 这里所提到的 OpenJDK JDK 7 是目前正在进行的 Java 7 规范项目, 由 Oracle/Sun 推动。OpenJDK 之前称为 Sun JDK, 是由 JCP 定义的规范, 是 Java 实现的官方参考。

在之前公开的日程表中, JDK 7 功能完成版是在 6 月 3 日完成, 最终的里程碑版本是在 9 月 9 日发布(正好赶在由 Oracle 举办的 JavaOne 大会之前)。如果我们相信这份安排, JDK 7 将在今年年底的某一天完成。但这是 JDK 7, 对于 Oracle 这样一家极其依赖企业级 Java 中间件的软件厂商而言, 在没有稳妥的解决 Java 7 规范的问题前, 我们很难看到它正式发布。

M6	2009/11/27 - 2010/02/18 (b84)	8 builds	Feature complete 51CTO.com 技术成就梦想
(ALL CHANGES)			
M7	2010/02/26 - 2010/04/15 (b89)	5 builds	
(ALL CHANGES)			
M8	2010/04/16 - 2010/06/03 (b96)	7 builds	
M9	2010/06/04 - 2010/07/22 (b103)	7 builds	
M10	2010/07/23 - 2010/09/09 (b110)	7 builds	

OpenJDK 上关于 JDK 7 的发布安排

没有标准的 Java, 没有“write once, run anywhere”。在 Java 7 规范方面, Oracle 不愿看到由其他 JDK/JVM 实现者提出的未定义功能; 其

他需要在 Java 平台上收获利益的厂商也无法忍受某一方对 Java 标准的独裁。这一切纷争或许会导致 Java 规范的第二次分裂危机。而一切喧闹的背后是默默注视和等待的开发者。

这一切就像一个咒语, 不断影响 Java 技术以一种偏离正确坐标的方向前进。

51CTO 精品杂志推荐《Linux 运维趋势》

《Linux 运维趋势》是由 51CTO 系统频道策划、针对 Linux/Unix 系统运维人员的一份电子杂志, 内容从基础的技巧心得、实际操作案例到中、高端的运维技术趋势与理念等均有覆盖。



本杂志长期处于探索期, 需要更多来自大家的意见与参与。如果您对这份电子杂志感兴趣, 那就请您下载阅读; 想要帮助我们做的更好, 请进入我们的 Linux 运维趋势技术圈留下您的宝贵意见建议。谢谢!

读者讨论组: <http://g.51cto.com/linuxops/>

邮件订阅入口:

<http://os.51cto.com/art/201011/233915.htm>

投稿邮箱: yangsai#51cto.com

发布周期: 每个月的第二个星期五

往期《Linux 运维趋势》下载汇总页:

<http://down.51cto.com/zt/71>

向Java开战?别搞错了对象

最近,我发现一个有趣的趋势,那就是在我身边的博客、DZone 和其他等等开发社区之中 Java 变得越来越不受欢迎。

作者 /Jonathan Fisher 译:贾隆

最近,我发现一个有趣的趋势,那就是在我身边的博客、DZone 和其他等等开发社区之中 Java 变得越来越不受欢迎。

只要你说到 Java 比起现在这些 web2.0 的宠儿语言们在某些地方还占些优势时,马上就会遭到这些语言的粉丝们的群起攻击,甚至会收到一些威胁言论。

在我看来,Java 广泛成功有着许多原因:

它被企业界所广泛接受;它是速度最快的语言之一;它是最安全的语言之一;内置在语言中的同步原语;它是平台独立的。

Hotspot 是开源的;成千上万家供应商提供了无数 Java 产品;成千上万个开源 Java 代码库;来自 JCP 的良好的社区管理(在 Oracle 之前)。

对于任何语言来说,这都是一个相当了不起的成绩,也展示了 Java 作为当下最流行的语言,在很长一段时间里无可匹敌的连胜纪录。

那么,为什么从 2010 年底到 2011 年,Java 突然间就变成了招人讨厌的魔鬼?

1. 现在反 Java 是一种时尚
2. 类似 C 语言的语法已经不再流行了
3. 人们把对 Oracle 的反感带到了对开发语言的兴趣当中
4. 人们接触到一些 Java 写的非常糟糕代码
- 5……此处插入另外一百个理由

Java 语言本身和它的 API 确实有不少实际问题,可以说举不胜举,想写也写不完的(比如混合本地和对象类型、一大堆被遗弃的 API、还有对受检查异常的不一致的使用等等)。但这里我是想伸出和平的橄榄枝……让我们讨论真正的问题,而不是把婴儿和洗澡水一起倒掉。

那么这个行业究竟出了什么问题?

Java 虽然有它的各种缺点,但确实已经完全征服了 Web 应用编程。而在场边,那些极富挑战性的新的编程语言正以目不暇接的速度一个接一个的出现,时刻挑战着人脑极限,它们同样也想要征服 Web 应用编程。这些新兴的语言们和 Java 一起竞争,它们就像赛车手一样蜂拥进站,或者就像街边的促销员正在为争夺街头领土而上演霹雳舞争霸赛。然而,就在大家争吵是 PHP 还是 Rails 3.1 哪个运行速度更快,哪个能够处理更多的同步请求,或者在我们幼稚的对语法和运行时的参数争吵不休时,总有一只沉默的大象在房间里暗暗的笑了。

那么告诉我,下面这些有什么共同点?

用信用卡买东西
去医院使用医疗卡
调整养老金
在牙医那里使用保险卡
寻找最好的汽车保险

向 Java 开战？别搞错了对象 II

银行之间的资金转移

医疗系统在药店配药

上面所有这些行业在美国的国家经济中都是上百亿美元的重头角色,而这些行业每天都在编写新的 COBOL 和大型机汇编程序。我没有添油加醋,我现在的工作就在配药行业,其他行业也曾经涉足过,或者采访过相关的人。

朋友们啊,老天作证,早在 1959 年就发明的 COBOL 在今天仍然几千行几千行的写着!我说的不是这儿几行那儿几行的那种维护工作,而是真正的全新的成千上万行,用来实现新的功能和新的要求,每一天都是如此。这些行业对新技术不闻不问,云计算里飘出来的风根本吹不到这些行业,但这些行业却是最必不可少的,它们构成了我们经济的基石。不仅如此,在这里,它们还从不创新,而且为了维护自己的传统技术一直进行着昂贵的大规模开销。

有多贵?我们简单谈一下大型机的许可费吧。比方说你的数据库买的是 MongoDB 企业版,只要把它安装好,然后你挂好 CPU,让数据库开始一个接一个的进行事务处理……下周你出去度假并留下 MongoDB 继续运行。等到你回来以后, MongoDB 并没有让你多花一分钱。

但大型机的软件许可就完全不同了。你要先花几百万买你的主机和它的数据库产品,再花了整整一星期才终于把 CPU 们挂上数据库的请求,然后你松了口气检查一下邮件,这时却发现早就收到了一份数据库供应商发来的 100 万美元的账单。等一下,我买了硬件,为什么还要付另一份账单?对了,大型机上的软件经常是按使用计费的,

或者按照你用了多少 CPU 周期。你在运行数据库时用了 2,000,000 个 CPU 周期,那你就要付给供应商 200 万美元。奇怪吗?就是这样。

这些行业就在你的生活中,你每天都在使用,却看不见摸不着,这里充斥着自大、老旧的系统和高成本。Java 一开始想要征服许多领域,虽然它征服了 Web 应用,但它在集中式计算上的努力最终还是告吹。这些大型机的行业完全到了应该降低成本和提高效率的时候了,但说实话,我们所取得的只能让自己尴尬。这些行业坚持使用那些老旧的系统,因为他们不认为 Ruby、Python、Scala、Lua、PHP 甚至 Java 能够处理这些行业的“负载”、可扩展性、或是满足他们那些老系统提供的上线时间的要求。这些理由完全站不住脚,而且在过去的 15 年间,这些行业“成功的”实现了零创新,而与此同时,网络技术一直在翻天覆地的跨越式进步。

因此,下周就会有人搞出另一个 DSL 让 Twitter 更容易使用,但你的银行还是会编出几万条全新的 COBOL 代码,让你在资金转账时再快上那么一点。我们为了 Java 是否过时争的面红耳赤,但整个国家经济都需要看到分布式计算的好处,如果友军的炮火继续误伤下去,那我们就都输了。停止这些荒谬的争论,把火炬和平的传递下去,试着征服几个庞然大物吧!

■ Java 编程开发小 TIP

在进行编程的时候最好首先建立一个空项目,然后在这个空项目里,编辑自己的 Java 程序。这样在你学习 Java 一段时间后,你就拥有了一个属于你自己的包,这对你以后学习就会很方便。

相同中不同:Java程序员不应低看C#

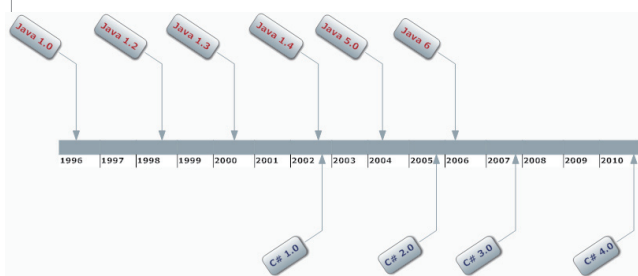
作者 / 贾隆

我们建立的开发网站 kalistick 算是个 Java 专营店了,从一开始的第一版起,全部产品使用的都是 Java 代码,但我们并不偏执,从 2008 年开始,我们也开始分析和使用 C#。

我注意到 Java 开发者大都在看待 C# 的时候多少带些轻蔑,就好像这是一只微软仿造的山寨猫,蠢人和软蛋们才会用它。我在博客中开始清除这些观念和误解,展示 C# 的种种好处。

Java 和 C# 的相同之处比不同处要多得多:两种语言都是写在大括号里的,就像 C 和 C++,类型都是静态、强类型和显式形态,两种语言都是基于类的面向对象语言,两者用的都是运行时编译的思路,并且很好的使用了垃圾处理。

所以这篇文章里,我要重点谈谈它们的相同点,以及 C# 的巧妙之处。



统一类型系统 (Unified type system)

在 Java 中,原始数据类型 (byte、int、bool、float、char 等) 和其他的类不同,它们并不算是面向对象,也不和引用类型共享相同的祖先类,但它们是有自己的包装类的,用来代表自己并且用来插入到对象结构中 (例如 int 使用 Integer 类),这样做可以提高性能。

在另一边, C# 的统一类型系统却都是从一个公用的根类型 System.Object 类中衍生而来的,即使是原始数据类型。所有的数据都要用到对象方法 (ToString、Equal、GetHashCode 等),所以你会碰上像 3.ToString() 这样的表达式,这种把方法混合到后缀,就带来了 dsl 风格的语句:

```
TimeSpan workingTime = 7.Hours() +  
30.Minutes();
```

这么做的美妙之处在于当开发者把数据类型当做值来使用时,它们能够和 Java 的原始类型一样高效,只有在想要把它们当做对象使用时,系统才需要使用 boxing/unboxing 来分配堆内存。

显式虚方法 (Explicit virtual method)

在 Java 中,默认所有的方法都是虚方法 (虽然这些方法可以使用 final 封装起来而不允许覆盖),而 C# 则不同,如果想在 C# 中写一个虚方法,必须先要用 virtual 关键字显式声明一下。

有几种原因决定了这样的选择,首先是性能上的考虑:虚方法都有一个悬在头上的性能问题,因为它们不是正常的内联,需要通过 vtable 来进行调用,这种做法并不直接 (Sun 的 JVM 可以内联上最经常调用的虚方法)。

第二个也是更重要的原因就是版本问题: C# 的设计思路是向后兼容,因此不同版本类库中的基类和衍生类是可以进化发展和保持兼容的。例如, C# 能够完全支持基类中新加入的成员和衍生类中的成员同名,而不会导致无法预料

相同中不同 :Java 程序员不应低看 C# II

的错误。最后一点是可读性:开发者的编程意图能够非常明显的读出来。在 Java 中,如果开发者不写出 Override annotation 的话,你不会知道他到底是不是想要重写这个方法。

```
class Meme { public virtual void Spread() {} }

class ThreeHundred : Meme { public
override void Spread()

{ Console.WriteLine("This is sparta!"); }}

class Dbz: Meme { // Not a method override
public void Spread() { Console.WriteLine("It's over nine
thousaaannnd!"); }}
```

真正的泛型 (True Generic)

关于泛型, Java 和 C# 显示出语法上的相似性,但真正深入理解之后你会发现这两种语言在泛型处理上的差别很大。

Java 的泛型是在编译器中处理的,运行时并不关心泛型类型。Java 在编译中使用叫做类型擦除转换的泛型类与方法:所有的泛型类型都被它们的原始版本替换,并且会在客户代码中插入 cast 和类型检查,生成的字节代码中并不包含任何泛型类型或参数的引用。Java 的泛型是让你在语法编写上尝到甜头,但不会让你的应用执行起来更有效。

而 C# 的泛型并不全是语言上的功能,它是放置在 CLR(Common Language Runtime, 相当于 JVM) 中的。在编译时需要进行泛型类型检查验证,但指定类型会推迟到类装载时生成。代码调用时的泛型是完全编译的,而且可以假设泛型在类型上是安全的,这被称为泛型的具体化。和 Java 不同, C# 不需要插入 cast 或者类型检查。泛型类型和方法可以通过引用 (class、delegate、

interface 等) 和 值 类 型 (primitive type、struct、enum 等) 来创建。

C# 中泛型的使用能够带来效率的提高 (不需要 cast 和值类型的 boxing/unboxing),还能够提高深层次的安全验证和反映能力。

```
public void AwesomeGenericMethod(T t)
where T : new() {

    T newInstance = new T (); // Causes a type
creation error in Java

    T[] array = new T [0]; // Causes generic array
creation error in Java

    T defaultValue = default(T);

    Type type = typeof(T);

    List list = new List (); } // Generic with same
name but a different number of generic type

public void AwesomeGenericMethod(T t, U u)
where T : new() {}
```

Oracle 的 Java 平台总架构师 Mark Reinhold 在 Devoxx 2011 大会上曾经探讨过给 Java 添加泛型的具体化问题,但这项功能还没有规划进 Java 的下一个主要版本中。

告别被检查异常 (checked exception)

Java 和 C# 的异常检查工作差不多一样,二者唯一的主要区别是:Java 中包含了 checked exception 这样的异常。在 Java 里你可以在方法声明中抛出 ExceptionType,这样做可以强迫任何调用方法的函数来处理异常。这个想法在纸面上说说很好,但实际使用中却很烦人,而且带来了新问题。

版本问题: 在新版本的方法声明中加入一个 checked exception 会破坏客户代码,就像给一个

相同中不同 :Java 程序员不应低看 C# III

接口添加方法一样。比如在版本 1 中你创建了一个 foo 方法,声明抛出异常 A 和 B,在版本 2 中你添加了一些新功能,抛出异常 D,这就是一个破坏性变化,因为现有的调用程序不能处理这个异常。

扩展性问题:在大规模的应用项目中,相互依赖的工作是非常多的,因此抛出的异常会多的难以统计,开发者经常会绕开掉这个功能,通过抛出泛型异常或者使用空的 catch 块。

checked exception 背后的思路是了不起的,但是尤其在大项目中,它有点太强迫性了。这就是 C# 为什么不使用 checked exception 的原因,其他主流语言也一样:留给开发者自己处理。

访问器和修改器

Java 的访问器和修改器 (getAddress、setAddress、isValid 等) 使用命名惯例。而在 C# 中,访问器和修改器是内置的,自身带有属性,不需要再写 getter 和 setter,所有的工作看上去都是直来直去,即使内部并不是这样的机制(许多其他语言也是这样)。

```
class Meme { // A private backing field is
    created by the compiler

    public string CatchPhrase { get; set;}

    public string URL { get; set;}}

    Meme meme = new Meme();

    meme.CatchPhrase = "Rick roll'd";

    meme.URL = "http://www.youtube.com/
    watch?v=EK2tWVj6lXw"; // Equivalent in Java
    class

    Meme { private String catchPhrase;
    private String url; public String getCatchPhrase()
```

```
{ return catchPhrase; }

    public void setCatchPhrase(String
    catchPhrase) { this.catchPhrase =
    catchPhrase; }

    public String getUrl() { return url; }

    public void setUrl(String url) { this.url = url; }}
```

当你声明一个属性为自动执行时,编译器会创建一个私有的匿名域,只有这个属性的 get 和 set 访问器可以读取。

这带来了兼容性,即使是在类的内部这个域也总是通过访问器使用,这看上去干净简练。

C# 中有一类访问器是 Java 中没有的:索引器 (indexer),它就像带有参数的 get 和 set。C# 中的 collection 类比如 Dictionary(和 Java Map 相类似) 使用 indexer。

```
var keywordsMapping = new
Dictionary<string, string>();

keywordsMapping["super"] = "base"; keywor
dsMapping["boolean"] = "bool";

Console.WriteLine("Java => C# : {0} => {1}",
"super", keywordsMapping["super"]);
```

你可能会说,没问题吧,这不就是写了一个初始化函数吗?

因为经常要创建对象,然后初始化,这些可以用构造器来完成,要不然在创建对象之后你就要调用不同的 set 方法。而对象的索引器可以在创建对象时就把值分配给对象的各种可以访问的域或属性,这样就不需要调用构造器了。

```
Meme leeroy = new Meme {
    CatchPhrase = "Leeroy Jenkins", URL = "http://
    www.youtube.com/watch?v=LkCNJRfSZBU";
```

相同中不同 :Java 程序员不应低看 C# IV

在 collection 类中也可以使用。

```
List<int> digits = new List<int> { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

Dictionary<string, string>
keywordsMapping = new Dictionary<string,
string>() { {"super", "base"},
{"boolean", "bool"}, {"import", "using"}};
```

逐字字符串 (Verbatim string)

从字符串中把字符分解出来是非常痛苦的工作,尤其是混合着不同含义的正则表达式。C# 的逐字字符串允许反斜杠、制表符、引号和换行符作为字符串的一部分,不再需要转义字符。

```
string pattern = @"d{3}-d{3}-d{4}"; string
input = @"Multiline string
325-532-4521"; Regex.IsMatch(input,
pattern, RegexOptions.Multiline);
```

总结

通过本文我想说 C# 不仅和 Java 很相像,而且它能够让开发者的生活变得更轻松,能够实在的减轻他们的负担(其他语言也一样),即使这是一只山寨猫,那么它做的也是相当不错。

实际上 Java 开发者们也做出了相似的回应,有些运行在 JVM 上的脚本语言例如 Groovy 就提供了这里说到的大多数功能,但 Java 本身还略显顽固,没有做出改变。

原文链接: <http://developer.51cto.com/art/201105/263914.htm>

Windows 8 将 .NET 和 Silverlight 打入冷宫?

在 D9 大会上,微软演示了 Windows 8 如何

为应用程序整合了一个以触摸操作为主的用户界面,这些应用都是基于 HTML5 和 JavaScript 开发的。这引起了 Silverlight 和 .NET 开发人员的担忧。

微软似乎将 Windows 8 的关注点都放在了新的 Web 标准开发平台,并用它来驱动全新的基于卡片(Tile)的触摸界面,却全然忽略了现有的 Windows 开发平台: Silverlight 和 .NET。

IDC 分析师指出,微软“没有指明 .Net 能在 Windows 8 世界的哪个角落立足”,微软原本应当在这方面为开发人员提供清晰的战略导向,但是却没有。

在演示 Windows 8 的应用程序时,微软甚至还表示:“这个应用程序是使用我们的新开发平台编写的,该开发平台基于 HTML5 和 JavaScript,人们可以使用他们已经在互联网上广泛采用的技术为 Windows 开发新的应用程序。”

如今已经有很多开发人员在微软 Silverlight 论坛上发表了愤怒的留言,并且表达了他们对微软 Windows 开发平台计划的担忧。

当前,微软并没有就此发表任何评论。Silverlight 和 .NET 真的要失宠了吗?

■ C# 编程开发小 TIP

委托是一种引用方法的类型。一旦为委托分配了方法,委托将与该方法具有完全相同的行为。

C# 委托具有以下特点:

委托类似 C++ 函数指针,但它是类型安全。

委托允许将方法作为参数进行传递。

委托可用于定义回调方法。

委托可以链接在一起。例如,可以对一个事件调用多个方法。

新手软件项目经理：你知道的太多了

■ 编者按
国人很喜欢评论，评论自己的工作，评论自己的上司。但是有的时候，对于一个项目经理来说，知道的太多了也不是一件好事。

话说有一天，魔鬼抓到了两个项目经理张三和李四，他决定吃掉一个，剩一个来给他做项目。张三骄傲地说：“我有 PMP 证书，我精通 CMMI、软件工程，还会敏捷、Scrum。”魔鬼一口就把他吃掉了，“额，你知道的太多了。”

1. 你知道的太多了

有 3 个游泳教练在教人游泳。第一个教练推荐学生看两本书《游泳工程》和《游泳理论大全 YYBOK》，看完了考试，考完试颁发《游泳证书》，可以自由下水游泳了，注意教练不对结果负责。第二个教练会先让学生在岸上按照《游泳 YYMMI》规范把动作练到完美，同时结合两本书《游泳工程》和《游泳理论大全 YYBOK》，并且在学生下水后严格按照《游泳 YYMMI》执行，不能变形。第三个教练很残酷，让学生在岸上练习了一会动作，就让学生带着浮板下了水。如果学生想靠边，教练会把他打到池中去。在一到两周时间内，学生就学会了游泳，但学生完全不知道啥是《游泳工程》，啥是《游泳理论大全 YYBOK》。对于继续深造的学生，教练才开始《游泳工程》和《游泳理论大全 YYBOK》的讲解，并且每讲一部分就让学生在游泳中多体会纠正。（还有一种，无师自通的狗刨。）

相信大家都很熟悉上面的内容，哈哈。有一个行业一直是第一种和第二种教法，教出了不少水鬼，这就是软

件行业。也许他们对《游泳工程》和《游泳理论大全 YYBOK》的理解非常到位，也许他们的动作完美符合《游泳 YYMMI》规范，他们甚至游得很快，唯一的区别就是他们再也不需要换气了。（我真该庆幸我女儿是跟第三个教练学游泳。）

“如果你的世界中只剩一种工具锤子，你会不会试图把所有东西都变成钉子？”虽然很想说不，但是对大多数人来说，答案是会。学到更多的知识有时是一种负担，一旦被洗脑成功，满地找钉子的人大有人在。

2. 其实只有两种项目管理

其实只有两种项目管理，一种是预言式项目管理，另一种是适应性项目管理。在软件项目中，当前预言式项目管理在行动上占上风，适应性项目管理在话语权上占上风。

2.1 预言式项目管理

预言式项目管理相信项目是可以清楚了解并预测的。预言式项目管理注重前期的计划，并认为项目无非是计划的执行过程，并在执行过程中控制变数。预言式项目管理视变化为威胁，要求严格的变更管理。因为人是带来变化的主体，所以更是威胁。

在预言式项目管理中，人员变成了角色，交互变成了过程和文档。客户变成了需求，领导变成了报告，开发人员变成了人月。从此，项目中的一切均

可量化管理。

2.2 适应性项目管理

适应性项目管理相信项目是无法准确预测的,要不断适应变化,响应变化。因为人是变化的主体,所以适应性项目管理特别注重人的作用。适应性项目管理通常采用迭代和增量式工作方式来进行探索和校准。

3. 无责任评项目管理

在上一篇《13 给新手的建议》中,我建议新手项目经理找到至少一位同公司的项目管理导师。因为导师就是你们的教练和浮板,他能在出现意外的时候保证你的人身安全(别变水鬼),他能让你把注意力集中到你当前要练习的地方,而不是面面俱到。但是每一位新手项目经理都想学到更多项目管理知识。应该学哪些呢?其实我更推荐直接在项目中学,跟着导师学。

3.1 项目管理知识体系 PMBOK

项目管理知识体系 PMBOK 是美国项目管理学会(PMI, Project Management Institute)开发的一个关于项目的标准,是 PMP 认证

基础。它把项目管理划分为 9 个知识领域,即:范围管理,时间管理,成本管理,质量管理,人力资源管理,沟通管理,采购管理,风险管理和综合管理。

无责任乱评:

PMBOK 偏向于预言式项目管理,希望理清项目中稳定的那些因素,控制和限制不稳定因素,例如人。

另外,对新人来说,直接看 PMBOK 容易陷入局部的理解,而不了解项目的整体。所以 PMP 考试的要求是至少 3 年以上的项目管理经

验,虽然现在貌似交钱就可以考。

如果 PMP 证书对找工作有帮助,就学吧,但注意不要被洗脑了哦。如果已经有了不少经验,希望更加全面的对项目的各个方面有个了解,PMBOK 也是不错的参考资料。

3.2 软件工程

软件工程是一门研究用工程化方法构建和维护有效的、实用的和高质量的软件的学科。

无责任乱评:

软件工程界长期以来偏向于预言式项目管理,也是希望理清项目中稳定的那些因素,控制和限制不稳定因素,例如人。近期软件工程书籍上开始引入敏捷。(也有人认为敏捷本来就是软件工程的一部分。)

因为过于全面,对新人来说,知识量太大,不建议开始就细读。

软件工程的优点是比较全面,如果希望更加全面的对项目的各个方面有个了解,软件工程是不错的参考资料。

3.3 CMMI

CMMI 是一种过程改进方法,它的目的是帮助组织改进他们的绩效。CMMI 认证是软件企业的流行认证,一度被认为是“银弹”。

无责任乱评:

CMMI 的本质是用于检查组织的软件开发能力的,这一点没错。但是有一点有问题的是,CMMI 本身没有能力提供真正的改进,它的改进必须依赖于预言式项目管理或适应性项目管理。换言之,CMMI 就是医院的检验科,它能提供检验报告,但是看病治病你还是得找医生。所以相信 CMMI 能看病的人都被骗了。用 CMMI 就

像去大医院治病，会带来昂贵的检验报告和昂贵的医生诊治费用，大医院养活着很多医生。

另外 CMMI 带来了大量的检验和昂贵的检验成本，本身就会影响项目成败。如果一个人每天都花钱去医院检验，他就有三种风险，一是无力承担昂贵的检验费用，另一种是过量的 X 光检验会致癌，第三种是检验占用的时间成本太大，没时间锻炼。不如改善生活态度，积极锻炼身体，定期去检验为好。

CMMI 也许对公司拿项目，撑门面比较有意义。但是作为个人，尤其是新手，CMMI 基本毫无意义。因为你读不懂检验报告，读懂了你也不会看病治病。还是专心把项目做好来得实在。

3.4 RUP

Rational 统一过程(RUP)是 Rational 软件公司(现在 Rational 公司被 IBM 并购)创造的软件工程方法。在 2000 年左右，没用 RUP，你出门都不好意思给人打招呼，掉份儿。

无责任乱评：

RUP 是典型的预言式项目管理，它那繁琐的流程定义，事无巨细的交付文档，和众多的角色分工都深深的出卖了它。但是沾了迭代两个字的光，它堂而皇之的混入了适应性项目管理的阵营。

它的优点是比较全面，如果希望更加全面的对项目的各个方面有个了解，是不错的参考资料。

3.5 敏捷和 Scrum

敏捷软件开发又称敏捷开发，是一种从 1990 年代开始逐渐引起广泛关注的一些新型软件开发方法，是一种应对快速变化的需求的一种

软件开发能力。它们的具体名称、理念、过程、术语都不尽相同，相对于“非敏捷”，更强调程序员团队与业务专家之间的紧密协作、面对面的沟通(认为比书面的文档更有效)、频繁交付新的软件版本、紧凑而自我组织型的团队、能够很好地适应需求变化的代码编写和团队组织方法，也更注重做为软件开发中人的作用。

Scrum 是典型的敏捷软件开发方法学，目前在所有敏捷方法学中采用率最高。

无责任乱评：

敏捷是典型的适应性项目管理，Scrum 当然也是。最大的缺点是理论与哲学很不“科学”，包含了很多人性化的部分，可变性很大，难于把握。在这一点上，预言式项目管理有很大的优势。因此敏捷极其简单又非常复杂。

敏捷更接近传统工匠文化中的手艺传递，特别需要师傅的指引。自学的，无论学了多少理论，都很难有好结果，这也是推荐项目经理导师的重要原因。

本文作者：大卫张

原文链接：<http://developer.51cto.com/art/201105/265843.htm>

■ 开发小 TIP

构成一个优秀程序员的一些特征：

以人为本

对学习有很大的胃口

掌握问题规律的本领

有一点神经质

执着

■ 编者按

Membase 容易安装、操作,可以从单节点方便的扩展到集群,而且为 memcached(有线协议的兼容性)实现了即插即用功能,在应用方面为开发者和经营者提供了一个比较低的门槛。

开源NoSQL系统又发力, Membase 1.7发布

Membase 容易安装、操作,可以从单节点方便的扩展到集群,而且为 memcached(有线协议的兼容性)实现了即插即用功能,在应用方面为开发者和经营者提供了一个比较低的门槛。

做为缓存解决方案,Memcached 已经在不同类型的领域(特别是大容量的 Web 应用)有了广泛的使用,其中 Memcached 的部分基础代码被直接应用到了 Membase 服务器的前端。

特性:

自动将在线数据迁移到低延迟的存储介质的技术(内存,固态硬盘,磁盘);

可选的写操作——异步,同步(基于复制,持久化);

反向通道再平衡[未来考虑支持];

多线程低锁争用;

尽可能使用异步处理;

自动实现重复数据删除;

动态再平衡现有集群;

通过把数据复制到多个集群单元和支持快速失败转移来提供系统的高可用性。

除了常规的稳定性提升和 bug 修复外,该版本还包含以下新功能:

新增的使用期限和同步复制的命令;

大大提高了集群,复制和再平衡;

错误条件的基本告警和可用性升级;

重新设计更好的用户界面,更精细的监测。

【小编碎语】什么是 Membase ?

Membase 是 NoSQL 家族的一个新的重量级的成员。

Membase 是开源项目,源代码采用了 Apache2.0 的使用许可。

该项目托管在 GitHub.Sourcetarballs 上,目前可以下载 beta 版本的 Linux 二进制包。

该产品主要是由 NorthScale 的 memcached 核心团队成员开发完成,其中还包括 Zynga 和 NHN 这两个主要贡献者的工程师,这两个组织都是很大的在线游戏和社区网络空间的供应商。

通过兼容多种编程语言和框架,Membase 具备了很好的复用性。在安装和配置方面,Membase 提供了有效的图形化界面和编程接口,包括可配置的告警信息。

Membase 的目标是提供对外的线性扩展能力,包括为了增加集群容量,可以针对统一的节点进行复制。

另外,存储数据进行再分配仍然是必要的。

(本文作者:红薯)

原文链接: <http://database.51cto.com/art/201106/267888.htm>

■ 编者按

然而就在最近,我注意到一个新的里程碑到来了——PHP 框架的新时代。5 月 21 日我在荷兰 PHP 大会上讲话时谈到这个问题,引发的热烈讨论也让我感到非常有趣。本文中我想要说说我所认为的 PHP 目前处在什么状态,哪些地方有问题,还有众多新出现的框架会怎样改变它。

新里程碑到来 开启PHP框架新时代

我曾在不同的系统和项目中埋头度过了很多年,大部分时间都花在了 PHP 上。然而就在最近,我注意到一个新的里程碑到来了——PHP 框架的新时代。5 月 21 日我在荷兰 PHP 大会上讲话时谈到这个问题,引发的热烈讨论也让我感到非常有趣。在最近的日子里好像一切都在变化。本文中我想要说说我所认为的 PHP 目前处在什么状态,哪些地方有问题,还有众多新出现的框架会怎样改变它。

框架的诞生

6 年前出现的 CakePHP 大概是第一个 PHP 框架,在它以后,我们见到了大量的 PHP 框架。目前的数量……可能大约有一百万了,这些框架使用着不同的 MVC、DBAL 和模板实现,我喜欢它们,即使它们各有各的古怪,而且都仍然没有得到大规模的采用。

如果你想要上网找找那些以框架为基础开发的 PHP 开源项目,能够找到的只有数量很少的一些,这很令人惋惜。部分原因是这些开源项目有很多在发布时甚至还没有任何 PHP 框架存在,另有部分原因是想用 PHP 框架来开发确实需要学习相当一段时间。也就是说如果一个 PHP 项目如果想用框架开发就会增加不少学习难度,至少在大多数情况下。

不过框架们的出现已经确实开始改变了我

们开发 PHP 的方式。许多开发者喜欢在口头上声称他们懂得面向对象,但是当框架出来后,他们要被迫证明他们真正懂得 OOP 和它的工作方式。如今你要是再让人用 `mysql_query` 可能脸上会挨上一两拳,因为他们现在还需要用到 `mysql_real_escape_string`。

框架是怎么做的呢？

没有人真正知道 PHP 框架应该是什么样子的,它们应该有什么功能。

因此,人们是怎么把它们开发出来的呢?好的,他们要么是跟着其他语言(比如 RoR)的现有框架照猫画虎,要么是自己拍脑门。因为不存在什么经验可言,现如今大多数的框架都是传统设计的,每个人都知道有问题,但却没法解决。

PHP 开发者一贯的务实做法在这里给了很大的帮助——就像 PHP 作为一种语言在不停进化一样,PHP 框架也在不断的根据反馈而改变和成长。有几年里多数人都对已经拥有的功能感到满足了,但是如果你现在再回头看看 2007 年的 Zend Framework 1.0,和 1.11 相对比功能已经天差地别。因此,即使在今天,框架也正在迅速的发展来满足功能的需要。PHP 4 曾经得到所有 PHP 框架的支持。这导致了大量的过时代码,特别是 OOP 模式。试图支持旧版本的 PHP 让新功能执行起来过于复杂,修正错误也愈发艰难。

新里程碑到来 开启 PHP 框架的新时代 II

而且,越来越少的开发者想要面对着这些旧代码工作。

什么出问题了?

每用一次 PHP 魔术函数,都有一只猫要死于非命!“魔法杀戮”

首先,回到 PHP 魔术函数(`__get`、`__call` 等等这些)盛行的年代。初看上去它们没有什么错,但它们实际上是非常危险的。

它们让 API 变得模糊不清,不能自动处理,最重要的是它们很慢。对魔术函数来说,它们的用例就是让 PHP 被破解而做不想做的事情,它们有用处,但是会发生错误的事情。

SCOP——Static class oriented programming, 面向静态类编程,这是我发明的术语,用来描述大部分 PHP 代码。静态方法从很多方面来说都有问题,但更重要的是,如果一个类只作为静态方法的集合,那么它就离 OOP 越来越远了,它只是使用类来作为存放各种函数的容器,但是甚至有完整的框架就在做这个。

Zend Framework 在很长一段时间是我最喜欢的 PHP 框架(现在对于 PHP 5.2 仍然适用),但我和它之间的主要问题是它总是过于卖力的想成为一个组件库。而且其他框架和它走着相同的路——它们各写各的类库,但却不使用现有的库。这就搞得 PHP 有那么多独立的库,你想要使用就需要下载整个框架,臃肿的框架真令人头疼。

2011 年的新时代

为了改善这种情况,人们终于选择要做一些事情。主要是在 PHP 5.3 的基础上从头开始重写框架。建立新的标准,统一所有框架之间的接口,

扔掉所有的遗留问题。这些听起来容易,但只有通过这些事情,我们才能进入框架的新时代。

在 CakePHP 诞生之前我没有用过任何 PHP 框架,所以我打算把它算作一个里程碑(其实我有些怀疑 CakePHP 就是第一个 PHP 框架,当然你不会把 Drupal 当做框架的)。从 CakePHP 开始,至今六年过去了,这六年我称为第一个时代。2011 年标志着第二个时代,全新的变革终将发生,我们会迎来各种各样的版本发布和公告。

有趣的是,2011 年的 PHP 好像不再是 PHP 了,或者可以说不再只是 PHP 了。有了 Nginx 和 CouchDB 这样的新工具,恼人的 LAMP stack 用的越来越少了。如今集成功能和互操作性成为了关键因素。同样,PHP 5.3 更像是一个全新出笼的野兽,它能够给你强大功能,但是并没有真正的支持向后兼容。

Git 支持许多 PHP 框架,给我印象最深的是 Symfony,它们已经成功的吸引到大量的贡献者,而对比前几年,PHP 框架最近步伐又快了很多。

PHP 框架们在不停的完善着各种小修改。首先,所有的魔术函数现在已经不见了,显式定义随处可见。此外,人们更多的考虑使用小尺寸的核心,通过扩展和类库来附加功能,这是一个了不起的办法,更易于使用框架,减少内存占用。

性能是 PHP 框架面对的主要问题,大部分框架在新版本中都做了提升性能的安排。比如 Symfony,在前端做了改进用来帮助管理资产(JavaScript 和 CSS),在 PHP 端由于清除了魔术函数和清理陈旧代码,通过 PHP 5.3 得到了巨大的性能提高。

新里程碑到来 开启 PHP 框架的新时代 III

我们来解决它,好吗?

新功能

显然,所有新的语言功能都包含在其中了,例如大多数的框架已经开始支持命名空间,还有更早出现的 PSR-0 现在很好的集成到框架中,匿名函数也以各种各样的方式在框架中找到了它们的位置。

依赖注射容器 (Dependency injection container) 和 Annotation 接口是我想提及的两个,它们能够改变你的编程方式。在 Symfony 中我很喜欢使用它们,现在其他的框架也正在迎头赶上,开始把它们集成进去。这些结合上新的 PHP 功能,就可以创建出非常干净的微型框架应用。

我并不喜欢直接把 Java 的功能列表移植到 PHP 来,Java 工作方式不同 (而且需要 1GB 内存)。

我们看到它正一步步走来,但我有点担心,因为我知道 PHP 喜欢轻一点的系统,而不是复杂的对象。这些功能听起来很炫,但实际上它们带来的问题比解决的要多。

因此,什么时候?

Zend Framework 2.0 正在路上,但还需要一段时间。由于 ZF 有一个庞大的代码库,他们首先要做的是把它转换成命名空间代码。完成之后,还要开始重新构建原有功能和新功能。ZF 目前正在做 MVC 方面的工作,我希望今年年底最终版能够发布。

Lithium 会来的快些,它还在 dev 模式,但似乎已经很接近完成了。Lithium 是一个和常规完全不同的框架,所以很值得看看。最触动我的是它的 AOP 实现,显然它只支持 PHP 5.3,但它对

CouchDB 和 MongoDB 支持的相当不错。

在我看来 Symfony2 依然在领先群雄。目前它正在 Beta2 版,但最终版也只是几个月的事儿了。Symfony2 功能列表很难总结,所以值得去它的网站查看一下,我想特别说一个——捆绑 (Bundle) 功能。捆绑是一种方法,可以通过外部的组件集合来扩展应用结构,想想插件吧。

结论

我对当前 PHP 行业发生的事情感到极度兴奋,我相信这些一定会带来重大的成就。我们终于可以扔掉所有 (或者大多数) 的遗留问题,开始实施新的想法。从现在开始五年之内,我们都会像今天一样兴奋。

原文链接: <http://developer.51cto.com/art/201105/264272.htm>

延伸阅读: PHP 框架发展存四误区

到目前为止,PHP 类框架还都是止于模仿,当然也确实充分利用了 PHP 的一些特性,但真正在应用上,仍然显得做大项目不足,小项目多余的感觉,这也是 PHP 框架一般只作为 "学者" 使用。

- 1、把控制器写得过于强大,偏离了框架本质
- 2、思想上仍然按照 Java 的那种老的一套思路,更适合于开发 B/S 应用的企业管理软件,而与 Web 的思路有点偏离
- 3、View 的模式过于死板,很难用于商业应用
- 4、框架本身过于庞大,导致实际开发中,很多人对框架进行不同程度的精简从而严重不统一

终上所述: PHP 框架要真正做到通用、实用,还需要很长时间,也许真正成熟之后,我们会后台编译在 PHP 扩展里的真正高性能 PHP 框架。

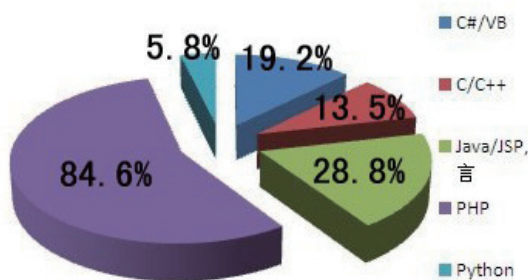
最受Web开发者欢迎的五大工具

工其事,必利于器。好的开发工具毋庸置疑会帮助 Web 前端开发者事半功倍,51CTO 在上期主办的技术沙龙《大型网站 PHP 开发之道》对现场的百余位 Web 开发者做了问卷调查,51CTO 调研小组分析得出了最受 Web 前端开发者欢迎的五大神器。

语言篇

PHP、Python 等 Web 专用的脚本语言,没有 Java、C++、C# 那么风光,毕竟面向公众的 Web 应用,在整个 IT 市场中未见得占有多大的份额。

但在五月份的 Tiobe 发布的新一期编程语言排行榜中,PHP、Python、Perl、Ruby 都还是给力的位居前十。由于来参加沙龙的百余名开发者大部分是做 WEB 前端开发,本次调查使用 PHP 语言的开发者比例达到了 84%,数据存在偏差,仅供参考。



图一：使用的语言的比例百分比

开源脚本语言 PHP

PHP,是英文超级文本预处理语言 Hypertext Preprocessor 的缩写。PHP 是一种 HTML 内嵌式的语言,是一种在服务器端执行的嵌入 HTML 文档的脚本语言,语言的风格有类似于 C 语言,被广泛的运用。

了解更多可参看:草根语言 PHP 的进化史

特点与技术优势

1. 跨平台特性: PHP 语言可以运行于 Linux、FreeBSD、OpenBSD、Solaris Unix、AIX、Darwin、Windows 等多种操作系统。

2. 数据库支持: PHP 支持多种主流与非主流的数据库,如: MySQL、SQL Server、Oracle 8PostgreSQL 等。

3. 安全性: 加密支持, PHP4 实现了完整的加密,这些加密功能是一个完整的 mycrypt 库,并且 PHP 4.0 支持哈希函数。Blowfish, TripleDES, MD5, 并且 SHA1 也是可使用的一些加密算法。使用 PHP 语言运行 Linux、FreeBSD、OpenBSD、Solaris Unix 操作系统,不需安装任何杀毒软件及补丁,安全可靠。

4. 扩展性: 扩充了 API 模块: PHP 4.0 为扩展的 API 模块的提供了扩展 PHP 接口模块,它比旧的 API 版本显著地快。

5. 执行速度快: PHP 是一种强大的 CGI 脚本语言,语法混合了 C、Java、Perl 和 PHP 式的新语法,执行网页比 CGI、Perl 和 ASP 更快,这是它的第一个突出的特点;内存方面, PHP 4.0 能更有效的使用内存,导致较少的内存占用消耗,这主要归功于引用计数技术的实现。

最受 Web 前端开发者欢迎的五大开发工具 II

6. 可移植性: PHP 写出来的 Web 后端 CGI 程序, 可以很轻易的移植到不同的操作系统上。例如, 先以 Linux 架的网站, 在系统负荷过高时, 可以快速地将整个系统移到 SUN 工作站上, 不用重新编译 CGI 程序。面对快速发展的 Internet, 这是长期规划的最好选择。

7. 功能全面性。PHP 包括图形处理、编码与解码、压缩文件处理、XML 解析、支持 HTTP 的身份认证、Cookie、POP3、SNMP 等。可以利用 PHP 连接包括 Oracle, MS-Access, Mysql 在内的大部分数据库。从对象式的设计、结构化的特性、数据库的处理、网络接口应用、安全编码机制等, PHP 几乎涵盖了所有网站的一切功能。

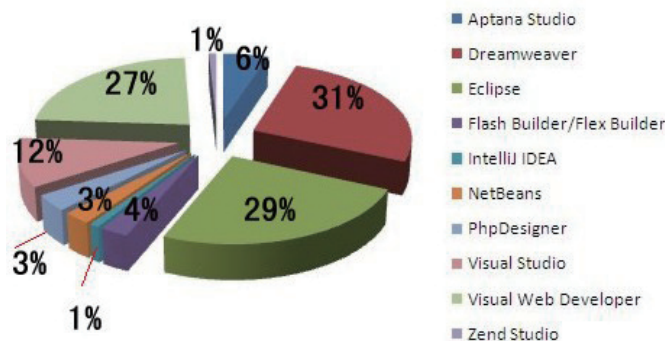
8. 可伸缩性。传统上网页的交互作用是通过 CGI 来实现的。CGI 程序的伸缩性不很理想, 因为它为每一个正在运行的 CGI 程序开一个独立进程。解决方法就是将经常用来编写 CGI 程序的语言的解释器编译进 web 服务器 (比如 mod_perl, JSP), PHP 就可以以这种方式安装。内嵌的 PHP 可以具有更高的可伸缩性。

9. 要发挥 PHP 语言优势, Linux+Apache+MySQL+PHP. 被称为“黄金组合”。

集中开发环境篇

集成开发环境 (IDE) 是用于提供程序开发环境的应用程序, 一般包括代码编辑器、编译器、调试器和图形用户界面工具。集成了代码编写功能、分析功能、编译功能、调试功能等一体化的开发软件服务套。本次调查开发者使用最多的 IDE 是

Dreamweaver. 其次是 Phpdesigner。



图二：使用的集成开发环境的比例百分比

Dreamweaver

Dreamweaver 是美国 MACROMEDIA 公司开发的集网页制作和管理网站于一身的所见即所得网页编辑器, 它是第一套针对专业网页设计师特别发展的视觉化网页开发工具, 利用它可以轻而易举地制作出跨越平台限制和跨越浏览器限制的充满动感的网页。它的优点就是直观性, 使用方便, 容易上手, 您在所见即所得网页编辑器进行网页制作和在 WORD 中进行文本编辑不会感到有什么区别, 但它同时也存在着致命的弱点: 一、难以精确达到与浏览器完全一致的显示效果。二、页面原始代码的难以控制性。所以对高层次的开发者不建议使用所见即所得网页编辑器, 可采用操作性也比较简单的 Phpdesigner。

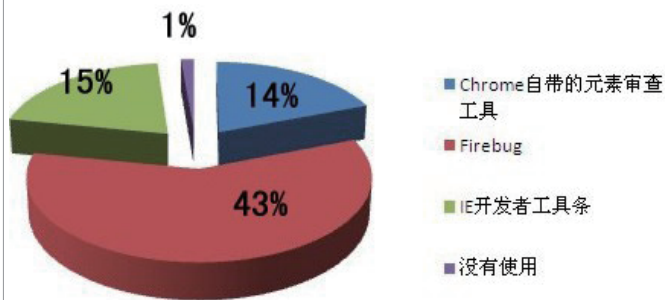
Phpdesigner

Phpdesigner 不仅支持 php, 也支持其他网络语言像 html, xhtml, xml, css 和 javascript, vbscript, java, c#, perl, python 等! Phpdesigner 7 支持 PHP 中你最喜欢的框架, 它也支持任何流行的 JavaScript 框架, 如 jQuery 的全雅虎 UI 库, ExtJS 原型和 MooTools。Phpdesigner 不像 Eclipse 那么麻烦, 不像 Dreamweaver 那么大。

最受 Web 前端开发者欢迎的五大开发工具 III

调试篇

网站开发者的一大苦恼,就是有太多的浏览器要应付。主流的调试工具: Chrome 自带的元素审查工具、Firebug、IE 开发者工具条。从下图我们可以看出在 Firefox 上开发和调试 web 站点是开发者的不二之选。



Firebug

Firebug 是 Firefox 上数一数二的好扩展,可作为 web developer toolbar 的补充工具。你可以用 Firebug 在线检查、监控、编辑网页上的 HTML、CSS、JavaScript 代码。他还提供一个脚本控制台,让 JS 编辑得更简单。控制台对象包含了一大堆选项,允许你输出代码到控制台然后进行调试。Michael Sync 提供一个更高级的操作指南,它可以知道你详尽地设置 Firebug 的选项。另外,Firebug 上有一个带宽监视器,能让你清楚地看到某些代码占了多少带宽。

HTML 玉不掩瑕

HTML 即超文本标记语言或超文本链接标示语言,是目前网络上应用最为广泛的语言,也是构成网页文档的主要语言。大多数 Web 开发产品采用附加的标记来扩展 HTML 处理数据的能力,HTML 现在最主要的问题是暂时还没有统一的标准,对动态数据处理不好,但它的简易型、

可扩展性、平台无关性,使得人们对它的喜爱忘记了它本身的缺点,从 W3C 在它身上所作出的改善就可以看到,HTML5 势必会成为新一代的 Web 语义标准。关于 HTML 5 可查看

51CTO 推荐专题: 链接: <http://developer.51cto.com/art/200907/133407.htm>

WEB 开发小 TIP

Web 设计师应向肖像画家学习 11 个经验

1. 工作迅速 (Work Quickly)
2. 不纠结于细节 (Don't Sweat the Small Stuff)
3. 熟识创作工具 (Know Your Tools)
4. 讲述一个故事 (Tell a Story)
5. 寻找内涵 (Tell the Story that Isn't There)
6. 无需另辟蹊径 (Don't Reinvent the Wheel)
7. 将客户摆在重要位置 (Your Client is the Subject)
8. Avoid Trends
9. 不忽视丑陋的设计部分 (Don't Avoid the Ugly Parts)
10. 花时间思考 (Take a Moment to Reflect)
11. 最后的百分之十 (The Last Ten Percent)

结语: Web 设计如同画肖像画,完成的作品越多,你速度会越快。锻炼自己,形成一个好习惯,便能助你简化进程,提高工作效率。

但比起这更值得做的是提高你的设计能力,保持开放型的学习经验,从而获得意想不到的效果。

■ 编者按

本文为 51CTO 独家译文,讲述了外国开发者 Genadiy Shteyman 从 Java 开发转向 Objective-C 需要掌握技能。文章中作者使用一个社交网络应用作为例子,演示怎样用这两种语言建立开发环境。

当Java遇见了Objective-C

目前在移动开发领域最重要的两个平台分别为 Android 平台和 iOS,在两个平台开发应用分别要用 Java 和 Objective-C 语言。

虽然 Java 和 Objective-C 就像是处在两个不同的世界,但这两种编程语言以及它们的平台库等等还是有许多相似的地方。

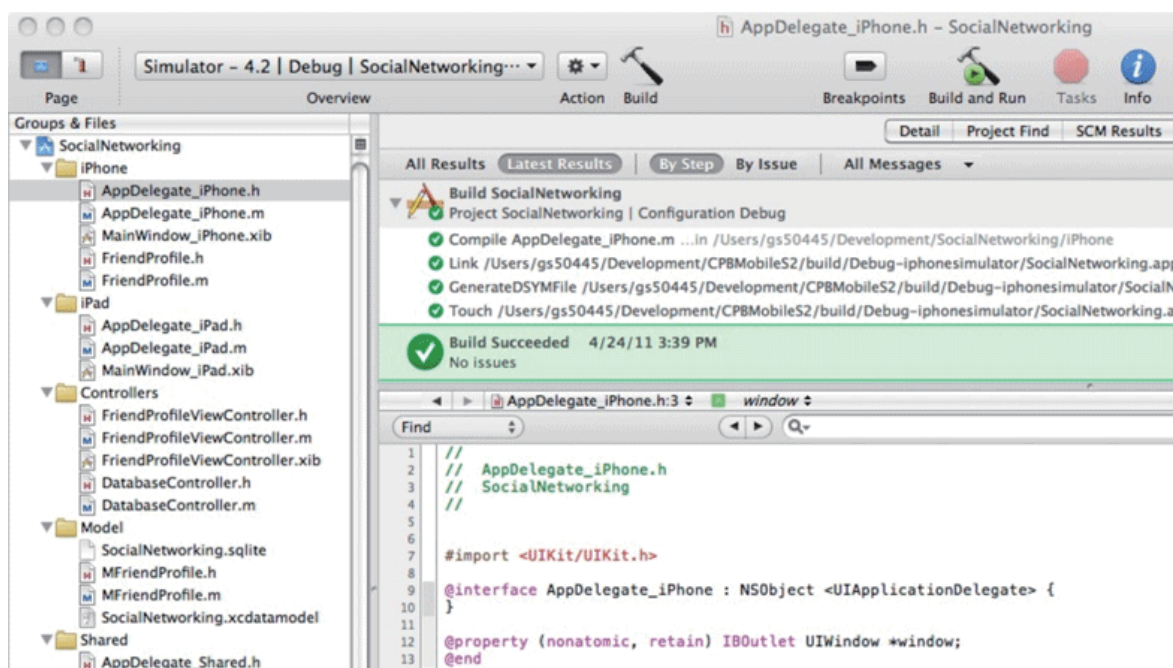
以下为全部译文,(文章中的“我”指代"Genadiy Shteyman"):

最近一段时间,我从编写企业 Java 应用转向使用 Objective-C。

经过长时间的困扰之后,我发现两者的相似之处很多,如果能够早些读到相关的文章,转换工作会容易得多。

所以我写下这篇文章,想要帮助 Java 程序员快速的掌握 Objective-C 开发的主要特点。我使用一个社交网络应用作为例子,演示怎样用这两种语言建立开发环境。

例子中会包括演示两种语言中数据的存储和获取。



图表一: Xcode IDE开发环境, 项目视图

当 Java 遇见了 Objective-C II

Objective-C 开发：从哪里开始

开发 iPhone 应用,首先最好要使用 Mac 电脑。最新的 Mac OS X 10.6 版本通常包含了一份 Xcode IDE,以及使用 Objective-C 的配套 iPhone 开发软件工具套装 (图表一)。

2010 年 11 月,苹果发布了期待已久的 iOS SDK 4.2,其中包含了丰富的框架和功能,用来搭建互动 iPhone 应用。

Xcode 还包含了一个仿真器,可以让你在电脑中模拟程序运行在手机上的效果。

Objective-C 是 iPhone 应用的主要开发语言。对 Java 开发者来说,幸运的是 Objective-C 是完全面向对象的,使用和其他 OO 语言相同的理念——继承、多态和封装等等。定义一个类 (Objective-C 中称为 module 或 .m 文件),首先要定义一个接口 (一个 header 或 .h 文件),然后把它引入到类中。

我们来看这个社交网络应用的例子,这个应用需要建立一个联系册,让你和朋友们时常保持联系。朋友的档案存储在 FriendProfile 对象中,包含四个字段:朋友的名字、城市、国家和电话号码,如 Listing One 所示:

Listing One

```
// FriendProfile.h

#import <Foundation/Foundation.h>

#import <UIKit/UIKit.h>

@interface FriendProfile : NSObject {}

@property (nonatomic, retain) NSString *
name;

@property (nonatomic, retain) NSString *
```

```
country;

@property (nonatomic, retain) NSString *
city;

@property (nonatomic, retain) NSString *
phoneNbr;

@end

//FriendProfile.m

#import "FriendProfile.h"

@implementation FriendProfile

@synthesize name;

@synthesize country;

@synthesize city;

@synthesize phoneNbr;

@end
```

在这个例子中,接口 FriendProfile: NSObject 表示我们定义了一个叫做 FriendProfile 的接口,它从 NSObject 基类中继承各种功能。NSObject 是 Objective-C 的根类,大多数 Objective-C 中用到的类都会从中继承,这和 Java 中的 Object 类相似。接下来,我们分配多个 NSString 类型变量 (等同于 Java 中的 String 类型) 用来存储朋友的数据。然后是建立 FriendProfile 类,使用 @synthesize 关键字自动创建各种 get 和 set 方法。建立一个 FriendProfile 对象可以使用如下的语句:

这里的 alloc 和 init 就像 Java 里的 new 关键字,用来在内存中建立 FriendProfile 对象。接下来,就可以给对象的各种字段赋值了。

模型在装载视图时开始启动。每个控制器都有一些从父类 UIViewController 继承而来的生命周期方法。比如 ViewDidLoad 方法就是其中

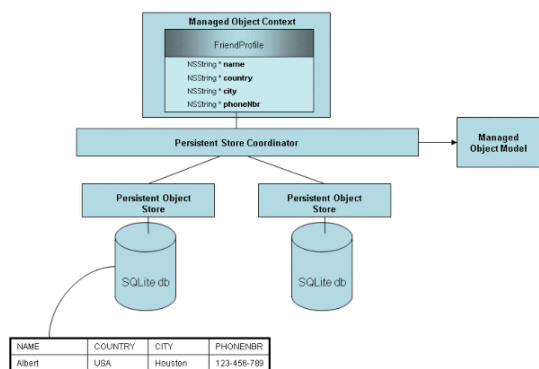
当 Java 遇见了 Objective-C III

之一,它负责在视图装载之后的额外设置,从数据库中取出信息,更新视图。在最简单的情况下,我们的视图包含一系列标签,或者是 UILabel 类型的对象,可以在应用运行时设置各种文本,用户可以立即看见朋友信息被更新了。

Objective-C 的数据库访问

较复杂的应用都会用到某类数据存储方式,通常是一个数据库。苹果推荐开发者使用称为 Core Data 的 Cocoa API 框架进行数据库存取操作。Core Data 框架能够直接与 SQLite 数据库相结合(我们例子中的数据库运行在移动设备上)。Core Data 隐藏了复杂的 SQL 操作,取而代之的是非常方便的 NSManagedObject 界面,你可以直接操作整个对象实例的各种字段,这些字段可以自动存入数据库。

Core Data 框架的另一个方便之处是在数据库中创建表(以及向表中添加关联与限制),这些都可以在 Core Data 的用户界面中完成。



图表二 Core Data stack 结

现在回到我们的社交网络应用例子,看看怎么从数据库中取出朋友的信息。我们使用 SQLite 和 Core Data API,但首先我们要稍微修改下 FriendProfile 类,代码请见 Listing Six:

Listing Six

```
//FriendProfile.h interface file// MFriendProfile.h

#import <Foundation/Foundation.h>

#import <CoreData/CoreData.h>

@interface MFriendProfile :
NSManagedObject {

}

@property (nonatomic, retain) NSString * name;
@property (nonatomic, retain) NSString * country;
@property (nonatomic, retain) NSString * city;
@property (nonatomic, retain) NSString *
phoneNbr;

@end

// MFriendProfile.m

#import "MFriendProfile.h"

@implementation MFriendProfile

@dynamic name;
@dynamic country;
@dynamic city;
@dynamic phoneNbr;

@end
```

Core Data 是一个非常有用的功能,可以让你通过图表来定义数据表和管理,可以动态生成相应的对象,而且无需使用复杂的 SQL 语句。

但不好的方面是这里有大量的没有经过封装的代码,这样你在使用它们与测试时需要非常小心。

Java: 数据库存取

Java 有许多数据库框架。在我看来, Hibernate 是和 Core Data API 最相像的 Java 框架。

当 Java 遇见了 Objective-C IV

Hibernate 使用的是对象关系映射 (Object-Relational Mapping, ORM) 机制, 这样你可以通过简单的在对象中设置字段并且直接映射成数据库中的表来把对象数据放入关系型数据库中。映射可以通过 XML 文件, 也可以通过 Java 5 中的 metadata annotation 方法获得。Listing Eight 是使用 XML 进行映射的一个例子。

Listing Eight

此例中, Listing Two 中的 FriendProfile 对象被映射到数据库中的一个同名表, 这是一种传统的数据映射做法。对象的四个字段被直接映射到表中的四列, 通过映射, Hibernate 可以使用 SQL 语句来完成各种操作。

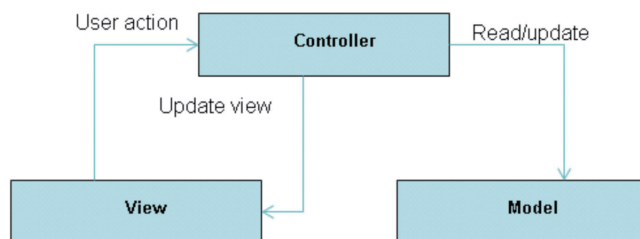
另一个配置文件叫做 hibernate.cfg.xml, 包含了数据库连接设置的详细信息, 包括数据库 URL、数据库驱动以及用户名和密码等, 代码请见 Listing Nine:

```
<hibernate-mapping>
  <class name="com.vo.FriendProfile" table="
FRIENDPROFILE">
    <property name="name">
      <column name="NAME" /> </property>
    <property name="country">
      <column name="COUNTRY"/>
    </property>
    <property name="city">
      <column name="CITY"/>
    </property>
    <property name="phoneNbr">
      <column name="PHONENBR"/>
    </property>
  </class></hibernate-mapping>
```

Listing Nine 中我们导入了所有需要的 Hibernate 库, 创建了一个 Hibernate Session 并且开始事务, 接下来我们仅简单使用了 Session 对象的 get 方法就轻松检索到了 FriendProfile 对象, 传递回所需要的对象类型并过滤出查询的字段——朋友的姓名。

Objective-C 的 MVC 模式

在 Java Web 应用中常采用 Model-View-Controller(MVC) 设计模式, iPhone 开发中也是如此。如果你在 iOS Reference Library 中查找 UIViewController 类的定义, 你会发现这样的话: “UIViewController 类为 iPhone 应用提供最基本的视图管理模型……你可以使用 UIViewController 实例来管理视图结构。” UIViewController 实际上是一个控制器组件, 用来触发业务逻辑, 更新客户端的视图。



Model-View-Controller(MVC) 设计模式

如果你想在 Xcode 中创建一个 UIViewController 类型的对象, 可以选择通过 XIB 文件来创建。这种特殊的 Xcode 文件定义了图形用户界面或者说视图, 包含了各种不同的控件, 比如按钮、图表和标签等等。

回到我们的例子中来, 假设你已经在联系列表中添加了几个朋友, 现在想按下某个朋友的链接来看查看他的详细信息, 这个功能可以通过定义控制器类来完成。

当 Java 遇见了 Objective-C V

Alloc 和 initWithNibName 都是控制器类创建实例时使用的方法,和 Java 的 new 关键字功能一样。

模型在装载视图时开始启动。每个控制器都有一些从父类 UIViewController 继承而来的生命周期方法。比如 viewDidLoad 方法就是其中之一,它负责在视图装载之后的额外设置,从数据库中取出信息,更新视图。在最简单的情况下,我们的视图包含一系列标签,或者是 UILabel 类型的对象,可以在应用运行时设置各种文本,用户可以立即看见朋友信息被更新了。

Java 的 MVC 模式

下面来看看如何使用 Java 后台在浏览器窗口中显示出朋友的详细信息。我们稍微修改一下 FriendlyServletController 即可,代码由于篇幅原因,请参见 51CTO 移动开发频道原文。

结论

除去语法结构与运行平台的不同,使用 Objective-C 进行 iPhone 开发与使用 Java 进行网络应用开发在下面几个方面是相同的:

- ◆两种语言都是面向对象的
- ◆两种语言使用同样的设计模式,如 MVC
- ◆两种语言使用相似的数据库存储技术,例如 ORM

然而,对于 Java 开发者,使用 Objective-C 时在有些地方要格外小心:

◆创建对象:Java 对象是在运行时通过 new 关键字创建的。因此 Java 程序员无需担心内存分配问题。而在 Objective-C 中,一个对象可以由三个关键字创建,alloc、new 或者 copy,这三个

关键字在创建对象时都会增加对象的持有计数(retain count),持有计数是 Objective-C 特有的内存管理方法,显示有多少个指针指向对象,是否可以被内存管理器回收。

◆销毁对象:由于强大的垃圾回收机制,Java 的内存管理工作极度简单。Java 的引用对象都存储在 JVM 的堆内存中,一旦不再被引用,就可以作为垃圾回收。Objective-C 使用的是内存管理器,而不是垃圾回收器。如果你使用上面说的三种方法在内存中创建了一个对象,那么必须使用 release 方法来释放对象。release 方法会减少持有计数,当计数降到 0 时,被引用的对象会接受一个来自高级类的 dealloc 方法,释放它占用的内存并重新分配。如果忘记了释放内存或释放失败,那么会造成内存泄露和不可预见的错误。

◆过多释放和过早重新分配内存:由于垃圾回收机制,Java 程序员可以完全不考虑这些问题。但 Objective-C 程序员需要小心,不能释放出比分配的更多的内存。如果在已经重新分配的对象上过多释放内存,就会造成应用的崩溃。

上面这些例子说明了 Objective-C 和 Java 在语法和语言元素上有很多相同之处。更重要的是,它们解决问题的思路 and 用到的组件也是非常相似的。如果你是 Java 程序员,相信你在看完这篇文章后,转向 Objective-C 的道路会更加通顺。

原文名称:Java Meets Objective-C

作者 Genadiy Shteyman

链接: <http://mobile.51cto.com/iphone-262962.htm>

请关注“Objective-C 编程语言 iOS 开发入门宝典”: <http://mobile.51cto.com/mobile/objc/>

Office应用 移动商务开发的新航标

2011年5月,51CTO记者走进了位于北京上地金山软件大厦,与金山移动办公部总监王晖一起聊了聊“Office软件移动化设计理念及市场前景”。



金山移动办公部总监王晖

究竟移动开发距离企业级开发有多远?成熟的PC应用如何移植到移动设备上?带着种种疑问,记者走进了位于北京上地金山软件大厦……

移动应用与企业办公

金山在移动开发方面所涉足的领域与企业级应用有着很深的联系。他们在移动开发方面的产品是大家熟悉的Office,只不过,在他们的努力下,平时只能坐在PC前操作的Office系统变成了“移动Office”。

“移动Office”这种全新的办公模式,可以让办公人员摆脱时间和空间的束缚。移动Office应用将使Office人员摆脱时间和场所局限,随时进行随身化的公司管理和沟通,帮助使用者有效地提高管理效率。在欧美一些国家已经形成一种Office的方式,而这种风潮也在中国流行开来。

金山移动办公部总监王晖介绍,根据可靠数据,今年包括iOS、Android的各个平台智能手机的总出货量第一季度已达到1亿台,保守估计今年全年智能手机的出货量将达到4亿台。经过系统调查预估,今年Kingsoft Office移动版将会有47万左右的用户。Kingsoft Office移动版的市场前景会非常乐观。

移动办公的迫切需求

对于Office应用在移动领域会不会受到用

户青睐这样的疑问王晖先生有着自己的看法:早期,计算机主要被科研部门所采用;发展到一定阶段后,计算机开始进入公司这样的机构,为人们办公使用;在经过一段发展,计算机开始进入寻常百姓家,计算机从纯粹办公用途向生活、娱乐方面发展。

但是,Office应用的比重还是很大的:移动设备上的Office应用或者说生产力工具的应用比重会变得很小,从最初的100%下降到20%,但是这只是相对份额。从绝对数量上来看,Office应用的需求量还是巨大的。王晖给我们举这样一个例子:“在日本,有些公司开始为维修工购买平板电脑,这个平板电脑上面就需要安装Office软件。原来维修工进行工作的时候需要拿维修手册等印刷品,但是当你有了平板电脑后公司只需要更新服务器上文档即可。当产品更新很快的时候,原来需要一年进行几次排版、运输等环节都可以用一次性购买移动设备来解决,成本将减少很多。也使以前不需要的人也将开始使用Office应用。”同样例子在欧美一些发达国家也有很多,尤其是医院医疗办公。医生可以拿着平板,在巡察病房的时候随时记录患者的病况,同时依靠3G网络或者WiFi同步把这些数据上传到服务器或者HIS系统等等。

Office 应用 移动商务开发的新航标 II

在这方面最著名的便是 BlackBerry Office 解决方案,比如最近热播的美剧《实习医生格蕾》第六季中就有西雅图格蕾思医院的医生们便拿着 BlackBerry 手机随时记录患者的情况。



16 个人的开发奇迹

而当记者询问金山在 Office 移动版投入多大的研发团队时,王晖的回答让记者非常惊讶,“只有 16 个人。”他介绍,整个 Kingsoft Office 移动版应用开始时只有查看引擎,其他什么都没有。尽管整个团队的交互设计设计师也只有一个人,但是为完成如此复杂的交互设计几乎所有人都成为了交互设计师。

交互设计在访谈中被王晖提及了数次,这也是把 PC 上应用移植到移动设备最大的难点,“Kingsoft Office 移动版这款适用于 Android 系统的应用与原来面向于 PC 版的 Office 相比在功能上相差还是非常多的,尤其是 UI 设计方面,简单地移植 PC 版的 Office 在触摸屏是没办法使用的,在 PC 上由于有鼠标键盘,但是在手机上按钮都挤在一起,这都需要重新设计。”随后王晖便向记者展示了 Kingsoft Office 移动版在 Android 平板和手机上是如何使用的。

在 PC 用户可以在文件夹中不断新建文件存放文件,但是在手机上只有一个 SD 卡,

Kingsoft Office 移动版为用户设计了所有文档功能。这一功能会检索到所有文件,用户可以在一个界面下像浏览照片一样浏览所有文件,而不用再去每一个手机文件夹中翻阅。

Kingsoft Office 移动版在交互设计中还有很多精彩的设计,如长按文档图标可弹出文档全名显示、页面设置等功能。而这一切都是来自 Kingsoft Office 移动版研发团队在不到一个季度的时间内完成的。

王晖向我们介绍:“Kingsoft Office 移动版研发团队在今年 2 月正式组建,由于金山在 WPS 上面研发人员紧张,研发移动版应用实际只有几个资深员工,我们只能再招聘一些新人,就是这样一个以老带新的团队,最终完成产品的基本应用却只用了一个季度稍多的时间。在六月份正式的 Kingsoft Office 移动版将与大家见面。”

相信很多移动应用开发者纠结于人员结构和时间上,而在开发中往往会束缚住开发者的手脚。当你了解要给用户什么样产品的时候,实践才是你唯一需要考虑的事情。

在路上的“Kingsoft Office 移动版”

记者在“Android Market”搜索“Office”时发现 1,340 款产品,但是中文简体的 Office 软件还是一个空白。作为中国 Office 软件的旗帜,金山很好把握这一契机,打造出自己在移动领域的新航标。对产品执着和专研显然也是每一个开发者必须的品质,希望每一位移动开发者也能打造出自己的经典应用。

原文地址: <http://mobile.51cto.com/hot-263061.htm>



51CTO技术沙龙 第六期

见微知著 大型网站PHP开发之道

PHP 是互联网站采用最多的语言之一,大家所熟知的 LAMP 构架基石就是 PHP。本次活动我们将请到两位大型网站的高级工程师,为大家分享他们所经历的 PHP 开发项目。并就这些项目的经验与大家探讨,究竟是什么决定了大型网站 PHP 项目的成败?

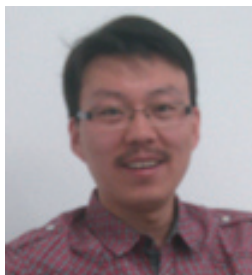
在 2011 年 5 月 21 日下午举办的第六期 51CTO 技术沙龙中,我们荣幸请到新浪乐居蒋宗君老师和隋龙飞老师。隋老师从基础为我们讲解了如何少走弯路做好 PHP 开发;蒋老师结合自身经验为我们讲解如何从 PHPer 草根成长到 Web 架构师。

PPT 下载地址:

关维家《少走弯路做 PHP 开发》<http://down.51cto.com/data/204409>

蒋宗君《从 PHPer 草根到 WEB 架构师》<http://down.51cto.com/data/204411>

视频专题 <http://developer.51cto.com/developer/51cto-salon-06/>



少走弯路做 PHP 开发

在线音乐网站技术总监 关维家

曾任中文在线高级工程师。一直在 sp, 图书领域工作, 现在从事音乐领域的研发工作。



从 PHPer 草根到 WEB 架构师

新浪乐居 蒋宗君

曾任搜狐网高级工程师,赛迪集团高级 PHP 工程师。目前正研究的领域: 结构化数据存储与数据中心, 非结构化数据的存储与分布式存储中心, 架构与研发。